AMENDMENTS TO THE SPECIFICATION

On page 6, please replace the two paragraphs comprising lines 4 and 5 as follows:

Fig. 7 shows a possible software architecture for a ~~Liaison~~ Content Provider unit 102~~104~~.

Fig. 8 shows a possible software architecture for a ~~Content Provider~~ Liaison unit 104~~102~~.

On page 8, please replace the paragraph comprising lines 23-25, as follows:

Fig. 36 shows how the content specifications and schedules that are produced by a Content ~~Receiver~~ Provider unit 102 can be used by the Content Filter 3508 of a Content Receiver unit 3402 as well as by the Insertion Engine 820 ~~720~~ of a Liaison ~~units~~ unit 104.

On page 9, please replace the paragraph comprising lines 12-20, as follows:

Fig. 40 shows how an UDP/IP packet can be packed into MPEG-2 Transport Stream packets using the known DSM-CC addressable section protocol. Each addressable section 4016 contains an addressable section header 4004, an UDP/IP packet 3908, and a CRC 4002~~)~~ (Cyclic Redundancy Checksum). The addressable section is then packed into MPEG-2 Transport Stream packets according to the MPEG-2 Systems standard (ISO/IEC 13818-1). Each Transport Stream packet contains a transport packet header 4010, and a portion of the addressable section in the transport stream packet payload 4008. The last packet in the sequence containing an addressable section may be only partially full, and the remaining space is filled with stuffing bytes 4006.

On page 11, please replace the paragraph comprising lines 13-24, as follows:

A software architecture for a Content Provider unit 102 is shown in fig. 7. The user interface 702 can be used to maintain information about Liaison units in a Liaison Store 704, and to maintain information about accounts and the bandwidth allocated to them in an Account Store 710, and to maintain specifications of content, including scheduling information, in a Catalog Store 706. Some content may be stored at the Content Provider unit in a Content Store 708. An interface 720 to Liaison units can be in the form of a CORBA client. This interface 720 can include an Exchange Account Info component 712 ~~714~~ that communicates with Liaison units about accounts and their bandwidth allocation profiles, a Send Catalogs component 714 ~~716~~ that sends content specifications and schedules to Liaison units, and a Send Content component 716 that delivers content to Liaison units. The Send Content component 716 may get the content from the local Content Store 708, or it may collect it from other locations 718 via FTP or some other suitable protocol.

On page 24, please replace the paragraph comprising lines 9-15, as follows:

A Liaison unit 104 can be implemented as software running on a Personal Computer (PC) running under the Windows NT version 4.0 operating system. The software can be written in the Java programming language. The PC can have an Ethernet adapter for the Liaison unit 104 to communicate with Content Provider units 102 via the CORBA IIOP protocol, with the Liaison unit 104 acting as a CORBA server. The Ethernet adapter can also be used by the Insertion Engine 820 of the Liaison unit 104 to send content to certain types of Mixing Units 108 (multiplexers and gateways) for insertion into the broadcast stream, using the TCP/IP or UDP/IP protocols.

On page 24, please replace the paragraph comprising lines 20-28, as follows:

The GUI for the Liaison unit <u>comprising an Account Management GUI 810 and a Configuration GUI 818</u> can be a Windows application with File, Edit and View menu buttons at the top of the screen in the usual way. An example of the File menu 2108 that could appear from clicking on the File button 2102 is shown in fig. 21. Typically only some choices would be available at any given time; others may be grayed out. An example of the View menu 2302 that could appear from clicking on the View button 2106 is shown in fig. 23. It could be used to switch to a functional area of interest, for example to maintaining information about Content Provider units and accounts (CP/Acct selection 2306), or to maintaining information about various configuration parameters (Config selection 2306). It could also be used to change the zoom factor for display of bandwidth allocation graphs (Zoom selection 2308).

On page 25, please replace the paragraph comprising lines 4-12, as follows:

This operation can have as input parameter an XML page that always includes the Content Provider ID, the account ID, and an add/delete/edit/info indicator. For an add request the page can include the account name, contact information, and a requested account allocation (list of interval allocations). For an edit request the page can include the requested account allocation. The operation can have as output parameter an XML page that includes the current bandwidth allocation for the account, and whether that allocation is unchanged since the last synchronization, or was changed by the user accepting a request, rejecting a request, accepting a request with modifications, or just making a unilateral change in the absence of any request. <u>The outputs of this operation can be stored in an Account Info storage 812.</u>

On page 25, please replace the paragraph comprising lines 14-17, as follows:

This operation can have as input parameter an XML page that includes the Content Provider ID. It can have as output parameter an XML page that includes a list of all account allocations for that Content Provider unit 102, with an indication of whether the allocation has changed since

the last synchronization, and if so why. The outputs of this operation can be stored in an Account

Info storage 812.

On page 25, please replace the paragraph comprising lines 19-23, as follows:

This operation can have as input parameter an XML page that includes the Content Provider unit

ID and a list of one or more accounts, with a list of one or more catalogs for each account. Each

catalog can contain a specification of content groups and content items, to replace any previous

specification for that catalog, or it can contain simply a "delete" attribute, indicating that the

catalog is to be deleted. The outputs of this operation can be stored in a Catalog Store 814.

Please replace the paragraph bridging pages 25 and 26, as follows:

This operation can have as input parameters the Content Provider ID, account ID, catalog

ID, and content item ID of the content to be delivered, plus the content itself in the form of an

octet string. (Note that this may not be a good choice for very large files or directories. It might

work better for the Content Provider unit to specify that the Liaison unit 104 fetch very large

files, rather than having them delivered.) If a directory is to be delivered, it can be delivered as a

single composite file, such as a "zip" file. The outputs of this operation can be stored in a

Content Store 816.

On page 26, please replace the paragraph comprising lines 3-7, as follows:

If a broadcast schedule appears in the catalog for the content, then the content can be put in the

Content Store 816, in a directory or database structure that supports using the account ID, catalog

ID and content ID as retrieval paths, so that the threads performing the actual broadcasting can

find them readily. If the content is identified in the catalog as "broadcast on receipt" then the files

can be put into the appropriate broadcast queue 3204 immediately.

On page 28, please replace the paragraph comprising lines 10-22, as follows:

Through the Accept Catalogs operation 804, the Liaison unit 104 gets a list of catalogs specifying the content items to be broadcast and the schedule for fetching them. The Liaison unit 104 can have two threads that handle the fetching 808. One of them can maintain a list of the content items to be fetched during the next 2 hours, say, sorted chronologically. It can update this list at appropriate intervals, perhaps once an hour, or whenever a new version of a catalog arrives or a catalog is deleted. (Developing the software to do this is tedious, but straightforward.) At the specified times the other thread can fetch the content items via some appropriate protocol, perhaps FTP. (The use of FTP rather than HTTP provides better support for fetching of directories.) Fetched items that have a broadcast schedule for them in the catalog can be put in the Content Store 816, in a directory or database structure that supports using the account ID, catalog ID and content ID as retrieval paths, so that the threads performing the actual broadcasting can find them readily. If the content is identified in the catalog as "broadcast on receipt" then the files can be put into the appropriate broadcast queue 3204 immediately.